Mil-Std-1553 Handling

Organization for Rack Monitors

Tue, Aug 19, 1997

For the D0 detector systems, the 1553 protocol is used heavily to support interfaces to Rack Monitors, from which the slow controls data is sourced. This document is a summary of how it was done.

Four modules of source code comprise the 1553 support, as follows:

Module	*Routine*	*Function*
`Enq1553`	`Cmds1553`	Data Access Table routine for queuing 1553 cmnds
	`OutW1553`	Output one word to RT via 1553 controller
	`InzQueue`	Initialize 1553 command queue on controller board
	`InzQ1553`	Initialize table of 1553 controller queue pointers.
`Int1553`	`Int1553`	Interrupt routine for 1553 controllers
`PP1553`	`PP1553`	Post-processing for completed 1553 commands
`Exec1553`	`Exec1553`	Execute single 1553 command, return error status.

The 1553 controllers are built as dual controllers on one VME board. The software can support up to sixteen 1553 controllers, although there are typically only 2 or 4 in one D0 local station; only one or two boards are used. These boards only decode A24 addressing, and each controller's memory allotment is 64K bytes. The VMEbus addresses used in D0 are `$200000` for controller #0, `$210000` for controller #1, etc. The software derives a controller# from bits 19–16 of the address. In each controller's 64K (non-volatile) memory, a queue is formed to support 1553 command activities. The queue is about 4K bytes in size and is located at `$F000` offset from the start of the controller's 64K memory space. This places the queue for controller #0 at `$20F000`, for controller #1 at `$21F000`, etc.

The 1553 Rack Monitor (RM), widely used in D0, provides interfaces for 64 A/D channels, 8 bytes of digital I/O and 8 D/A's. Each RM is accessed via the 1 MHz 1553 serial bus as a Remote Terminal (RT). The VMEbus-based local stations house 1553 controllers, each of which connects to a number of RM's. Each 15 Hz cycle, all A/D channels are read and all digital bytes are read. As an RT, up to 32 channels of A/D can be digitized and read with one command. Each of the four words of digital I/O is accessed as one word for each command. Accessing 32 channels of A/D in this way takes about 800 µ s, while reading each word of digital I/O takes about 125 µ s. The software is organized so that multiple 1553 controllers can be busy at the same time, with each controller reading data from the RM's to which it is connected.

Normal data acquisition from 1553 RM's is done via Data Access Table (DAT) entries (type $0C) each of which invokes the `Cmds1553` routine to queue commands to the individual 1553 controller's queue. As the first command is queued for a given controller, it is started immediately; but the logic does not wait for its completion in

order that more commands may be queued with subsequent DAT entries. The idea here is that it is more efficient to operate more than one controller at the same time; a suitable choice of DAT table entries makes this happen. When all such 1553 commands have been queued, a special DAT entry (type $0E) invokes the `PP1553` routine to await the completion of 1553 activity from all controllers that are busy and to perform do the post-processing that deposits the 1553 data read into the data pool. This allows subsequent DAT entries of other types to refer to the freshly-acquired 1553 data.

As a controller delivers an interrupt at the completion of each 1553 command it executes, `Int1553` is invoked, which records any error status in the command block. If another command is waiting in the queue, it is started right away. If no further command is waiting in that controller's queue, the routine simply exits. There is also provision for keeping some diagnostics about the activity of the command block in a structure following the command block proper. The offset to such a structure is found in the hi byte of the `errStatOrd` word, the first word of the block. (If this offset is zero, no such diagnostics are recorded.) At the offset indicated, measured from the start of the command block, are 8 bytes of diagnostics with the following structure:

Field	*Size*	*Function*
`errStat`	word	Last error status received from the addressed RT
`errCnt`	word	Count of errors detected with this command block
`execCnt`	long	Count of executions of this command block

Each time the command block is used, the `execCnt` is incremented. If there is an error, the `errCnt` word is incremented and the received status word is put into `errStat`. Separate from the RT status word, the controller's error status is placed in the hi nibble of the low byte of the `errStatOrd` word of the command block. The diagnostic data can easily be viewed in a memory display of the command block.

A 1553 command block has the following structure in the case that data is being *requested from* an RT:

Field	*Size*	*Function*
`errStatOrd`	word	Completion error status, controller order code
`command`	word	Command sent to RT: RT(5) T/R(1) SA(5) WC(5)
`status`	word	Status returned from addressed RT
`data`	≤ 32 words	Data returned from addressed RT

The 1553 command word has four fields. The upper 5 bits specify the address of the RT to be accessed. The next bit indicates the direction of data flow. If T/R=1, data is to be received from the RT; *i.e.,* the RT is commanded to transmit data back to the controller. The next field specifies the 5-bit Sub-Address that indicates to the RT

what part of its data is being accessed. The lower 5 bits is the Word Count field, in which a value of 0 is interpreted as meaning the maximum count of 32 words.

When a command is started, by writing the address of a command block to a register, the controller uses DMA to access the first word of the command block, where it finds an order code in the low 3 bits. The order code should have a value of 2, meaning that it should transmit the following command word to the 1553 serial bus, to which up to 30 RT's may be connected. (In D0, eight is a typical number of RT's connected to a single controller's bus.) Each RT has a distinct address, so that only one RT recognizes a match with its own address in the upper 5 bits of the command word it sees on the serial bus. The addressed RT must begin responding to the controller within 14 µ s, or the controller will assume there is no response and abort the command with error status. The response consists of a status word followed by the requested data words. (For the case of sending data to an RT, the controller transmits the data words immediately following the command word, which the addressed RT should receive, after which it responds with a status word that the controller places in the command block immediately following the last data word it had transmitted.)

For the case that data is *sent to* an RT, the 1553 command block has the following structure:

Field	*Size*	*Function*
`errStatOrd`	word	Completion error status, controller order code
`command`	word	Command sent to RT: RT(5) T/R(1) SA(5) WC(5)
`data`	≤ 32 words	Data returned from addressed RT
`status`	word	Status returned from addressed RT

Note that in this case, the data words reside in the command block beginning after the command word, and the RT's status response is recorded in the command block immediately following the last data word that was transmitted to the RT, so that the location of the status word therefore depends upon the number of data words transmitted.

Here is an example of a memory dump showing a command block that outputs a word to set the initial A/D channel# to be digitized, followed by a command block that reads and digitizes the 32 channels beginning with that channel#.

```
M MEMORY DUMP     08/18/97 1332
0731:200600 78A2 13F1 0020 1000
0731:200608 0000 0000 0000 0000
0731:200610 0000 0000 0000 0000
0731:200618 0000 0000 0000 0000
0731:200620 0000 0000 0000 0000
0731:200628 0000 0000 0000 0000
0731:200630 0000 0000 0000 0000
```

```
0731:200638 0000 0000 0000 0000
0731:200640 0000 0000 0000 0000
0731:200648 0000 0000 0000 0000
0731:200650 0000 0000 0000 0000
0731:200658 0000 0000 0000 0000
0731:200660 0000 0000 0000 0000
0731:200668 0000 0000 0000 0000
0731:200670 0000 0000 0000 0000
0731:200678 0753 07CB ABF2 A663

0731:200680 78A2 1680 1000 FC11
0731:200688 FCB2 FD13 FD74 FDB5
0731:200690 FDF6 FE37 FE78 FEF9
0731:200698 FF1A FF3B FF5C FF7D
0731:2006A0 FF7E FF9F FFD0 FF91
0731:2006A8 FFB2 FF93 FFB4 FF95
0731:2006B0 FFB6 FF97 FFF8 FFB9
0731:2006B8 FFBA FF9B FFBC FFBD
0731:2006C0 FFBE FFBF FFB0 0000
0731:2006C8 0000 0000 0000 0000
0731:2006D0 0000 0000 0000 0000
0731:2006D8 0000 0000 0000 0000
0731:2006E0 0000 0000 0000 0000
0731:2006E8 0000 0000 0000 0000
0731:2006F0 0000 0000 0000 0000
0731:2006F8 06D3 07C7 ABF2 A654
```

The first command block causes channel# 32 to be digitized. The second command block reads that channel and automatically causes the next channels to be digitized and returned, keeping pace with the 1553 serial rate of 20 µ s per word. Note the offset value of $78 to the diagnostic area found in the hi byte of the first word in each command block. The controller status is found in the third nibble of that same word and has the value $A in both cases, indicating "message complete" status without errors. (If an error occurs, this nibble normally has the value $E.) The order code is 2 in the low nibble, which indicates that the command block is valid. The second word is the command word itself. The hi 5 bits indicate the RT address of 2. The next bit indicates that the first block is an output (RT should receive data) and the second block is an input (RT should transmit data). The next 5 bits indicate the SA sub-address field that is a 31 in the first case and 20 in the second. The first case is a special part of the 1553 standard that means "mode code". For the Rack Monitor, it means set the initial A/D channel#. The second block's SA asks for A/D data to be digitized for consecutive channels beginning at the one set by the mode code, for as many as the NW Number of Words field indicates, 0=32 words in this case. (Much of the structure shown is unused. These command blocks are sized at 128 bytes just to make it convenient to view them in 64-byte chunks.)

Following the command word is the output data word of 0020 that sets the initial channel# of 32, followed by the status word returned from the RT. In the second command block, following the command block, is the status word returned by the RT followed by the 32 words of data. Because the RM supports 64 channels, there are

two such pairs of command blocks associated with each RM each of which collects 32 channels of A/D data. The total time required to read 64 channels is less than 2 ms.

The PP1553 task-level code can tell from the queue header whether a controller is busy, and it can monitor its progress as each command is completed. But PP1553 also does any post-processing required by the command. Post-processing refers to copying the data, placed by the controller into the 1553 command block associated with each command, into the data pool—for those commands marked to require such post-processing. The 8-byte 1553 command queue entry has the following structure:

Field	*size*	*Function*
cmdOff	word	Offset to command block (within 64K memory)
typSiz	word	4-bit type code, 2-bit retry count, 10-bit step size
ptrMem	long	Ptr to first destination table entry field

The type code values used as as follows:

1 Data Access Table processing via Cmds1553
2 OutW1553 routine (one-word settings)
3 Exec1553 routine

Post-processing is only done for DAT processing, as it is needed to copy the data from the 1553 command block memory into the data pool. To do that, it needs the pointer to the initial destination table entry field, plus the step size between successive table entries. For example, the ADATA table has 16-byte entries for each analog channel. The ptrMem would be set to the initial target channel's reading field, and the step size would be 16.

DAT queuing entries (type $0C) may produce one or more 1553 commands, according to the count field. The memory address field points to the first 1553 command block in the controller's memory, so that the controller# can be gleaned from this address. A step size field indicates the spacing of successive command blocks. A 1553 command block, as described above, is a structure that includes the command word itself, plus space for the data words and the RT status word for that command's transactions. The command blocks are built into the controller's (non-volatile) memory as part of system configuration. All command blocks in controller memory are usually laid out similarly across multiple controllers, since they target RM's. To maximize the overlap of multiple controller activity, one or two commands are queued to each controller, then the remaining commands are queued. This gets both controllers started almost immediately, as additional commands are queued. Then the controller activities will run interrupt-driven until all queued commands have been executed. The PP1553 routine, while it awaits controller inactivity, also monitors its progress, performing any post-processing it can as soon as each command completes. If a command block's execution failed, any data words that were to be copied into the data pool are

set to zero. This may produce alarm messages, but it is thought to be better than leaving the data pool values alone so that they might become stale. In the queue header in each controller's memory, there is a 32-bit array (4 bytes) that holds an error flag for each RT that is accessed. These bits may also be monitored for alarms so that one can get an alarm message that identifies the RT in error.

A Rack Monitor's I/O includes 64 channels of A/D, accessed via two 32-word 1553 commands, each of which is preceded by a one-word output that sets the initial channel for the digitization sequence, 0 or 32. (See above example.)

The retry-count field is only used for `OutW1553` queue entries. This is done to lessen the chance a D/A setting will fail. In the unlikely event that a setting resulted in error status due to noise, one or two retries may almost guarantee success.

One more routine, `Exec1553`, is used that is invoked by the 1553 Test Page Application. This subroutine issues one 1553 command of any type, waits for it to complete, then returns both the controller status and the RT's returned status. Using the Test Page, one can prepare up to 32 data words for output, or can view up to 32 data words received. One can prepare a list of such commands to be executed in sequence, monitoring error statistics for all. Note that one cannot operate more than one 1553 controller simultaneously with the Test Page. Only DAT processing can drive multiple controllers at the same time. For each command that is executed, it is first prepared in the controller's memory at a fixed offset of $0080 before execution.